

Rancang Bangun *Management Service Platform* (MENTOR) sebagai Pendukung Ekosistem IoT

Muhammad Baso Adrian Ibrahim¹, Kasim², Eddy Tungadi³

¹ Teknik Elektro, Politeknik Negeri Ujung Pandang
baso.adrian.25@gmail.com

² Teknik Elektro, Politeknik Negeri Ujung Pandang
kasim@poliupg.ac.id

³ Teknik Elektro, Politeknik Negeri Ujung Pandang
e_tungadi@yahoo.com

Abstrak

Ekosistem Internet of Things (IoT) berkembang dengan sangat cepat dan diperkirakan akan menghubungkan 5-20 miliar perangkat pada tahun 2020. Namun, dalam penerapan ekosistem IoT untuk manajemen data dan konfigurasi terkadang harus dikelola langsung dari perangkat yang ada di lapangan. Platform IoT merupakan sebuah platform layanan yang dapat mempermudah dalam manajemen perangkat IoT, manajemen keamanan, manajemen policy, dan lain-lain. Dari penjelasan sebelumnya, maka penelitian ini membuat sebuah platform IoT yang dalam hal ini diberi nama "*Management Service Platform* (MENTOR) Ekosistem IoT", sebagai alternatif yang mampu mendukung komunikasi berbagai perangkat IoT dan menjamin validitas dari perangkat yang mengirimkan data. Hasil pengujian menunjukkan bahwa platform MENTOR IoT mampu berjalan dengan berbagai protokol yang berbeda, melakukan kontrol perangkat dan dilihat dari banyaknya pengguna yang mengirimkan data secara bersamaan dan terintegrasi serta waktu yang dihasilkan saat mengirimkan data melalui akses lokal dan akses publik tidak jauh beda. Sehingga berdasarkan hasil tersebut menandakan bahwa platform ini sangat *reliable* dan *scalable*.

Keywords: *Platform IoT, Ekosistem IoT, Cloud IoT*

I. PENDAHULUAN

Pada saat ini, ekosistem IoT pada umumnya menggunakan model sistem terpusat yang menghubungkan berbagai perangkat dalam dunia fisik dengan berbagai protokol berbeda. Penerapan IoT menjadikan aktivitas dalam berbagai bidang dapat saling terhubung melalui Internet, serta menjadi lebih mudah dan efisien. Manajemen ekosistem IoT terdapat keterbatasan dalam hal komputasi dan penyimpanan karena menggunakan komponen penyimpanan dan komputasi yang terbatas.

Dalam penerapan ekosistem IoT juga biasanya untuk manajemen data dan konfigurasi terkadang harus dikelola langsung dari perangkat yang ada di lapangan. Keterbatasan pada perangkat IoT ini membuat penyimpanan dan pengolahan data yang besar serta manajemen perangkat perlu dialihkan ke sebuah platform IoT. Dengan adanya platform IoT pengelolaan data dapat diakses secara luas tanpa harus berada di lokasi perangkat. Maka dari itu, penelitian ini bertujuan merancang dan membangun sebuah platform IoT yang dapat memajemen berbagai *service* yang telah dibuat dan sebagai alternatif yang mampu mendukung komunikasi berbagai perangkat IoT dan menjamin validitas dari perangkat yang mengirimkan data.

II. KAJIAN LITERATUR

A. Platform IoT

Platform IoT adalah sebuah platform yang untuk mengelola komponen sistem lainnya dan menyediakan

Layanan Web dengan protokol komunikasi untuk pendaftaran dan kontrol sensor. Fungsi utama dari platform IoT sendiri yaitu mengoordinasikan komunikasi antara komponen-komponen lain dan menyediakan layanan manajemen sistem [1]. Menurut Singh dan Viniotis platform IoT merupakan sebuah platform layanan yang dapat mempermudah dalam manajemen perangkat IoT, manajemen keamanan, manajemen policy, dan lain-lain [2]. Terdapat beberapa platform IoT yang dapat kita ketahui seperti Xively, Nimbits, Thingspeak, Paraimpu, iDigi Device Cloud dan SensorCloud [3].

B. IoT Networks Protocols

IoT network protocols merupakan protokol yang digunakan untuk menghubungkan perangkat melalui jaringan dan merupakan seperangkat protokol komunikasi yang biasanya digunakan melalui Internet. IoT network protocols memperbolehkan dalam penggunaan komunikasi data end-to-end dalam lingkup jaringan. *IoT Network Protocols* yang biasanya digunakan yaitu HTTP, MQTT, CoAP, ZigBee, Bluetooth, dan LoRaWan [4].

C. Kubernetes

Kubernetes adalah sebuah sistem open source untuk mengoptimalkan penyebaran, scaling dan pengelolaan kontainer. pengelompokan kontainer yang terbentuk dalam logic unit atau yang disebut pod untuk mempermudah pengelolaan [5].

D. Mikrokontroler

Mikrokontroler adalah suatu chip berupa IC (Integrated Circuit) yang dapat menerima sinyal input, mengolahnya dan memberikan sinyal output sesuai dengan program yang diisikan ke dalamnya. Sinyal input mikrokontroler berasal dari sensor yang merupakan informasi dari lingkungan sedangkan sinyal output ditujukan kepada aktuator yang dapat memberikan efek ke lingkungan. Jadi secara sederhana mikrokontroler dapat diibaratkan sebagai otak dari suatu perangkat/produk yang mampu berinteraksi dengan lingkungan sekitarnya [6]. Salah satu jenis mikrokontroler seperti ditunjukkan pada Gambar 1.



Gambar 1. Mikrokontroler Arduino

E. Sensor

Sensor adalah suatu peralatan yang berfungsi untuk mendeteksi gejala-gejala atau sinyal-sinyal yang berasal dari perubahan suatu energi seperti energi listrik, energi fisika, energi kimia, energi biologi, energi mekanik dan sebagainya [7].

Sensor mengukur hampir semua aspek dunia fisik. Kalibrasi sensor memungkinkan mereka untuk disesuaikan dengan fungsi spesifik aplikasi, sehingga mereka dapat mengukur hal-hal seperti suhu, getaran, listrik, dan komposisi udara dengan akurat. Seringkali, data sensor ditugaskan untuk menangkap informasi yang relevan dengan tugas tertentu, sehingga data tersebut dapat digunakan untuk melakukan perbaikan proses dengan tujuan menghemat uang atau meningkatkan efisiensi [8].

F. Data Visualization

Visualisasi adalah penggunaan representasi data yang didukung komputer. Tidak seperti data statis visualisasi, visualisasi data interaktif memungkinkan pengguna untuk menentukan format yang digunakan dalam menampilkan data [9].

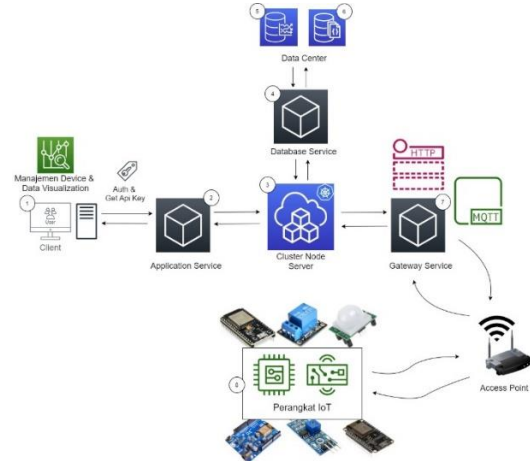
Untuk mendukung sensor IoT yang terus mengumpulkan data/informasi, diperlukan visualisasi data berupa dashboard pemantauan yang akurat [10]. Dalam visualisasi data, menurut Song, dkk Highchart sederhana dan nyaman untuk menambahkan grafik interaktif di situs web atau web aplikasi. Highcharts tidak perlu plug-in dan berjalan cepat karena dapat diperoleh dengan menggunakan AJAX serta data dapat dikonversi ke format data JSON[11].

III. METODE PENELITIAN

A. Arsitektur Sistem

Arsitektur sistem merupakan penggambaran umum untuk sistem yang akan dibuat. Pada Gambar 2

menunjukkan model perancangan arsitektur sistem yang dibuat berdasarkan dari hasil analisa kebutuhan.



Gambar 2. Arsitektur sistem

Berikut rincian penjelasan pada Gambar 2 :

1) Client / komputer

Client / komputer user digunakan untuk mengakses Platform MENTOR IoT dengan memonitor data dan perangkat

2) Application Service

Application Service merupakan service yang dibangun pada pada kubernetes yang digunakan untuk menangani aplikasi terhadap client.

3) Cluster Node

Cluster Node digunakan untuk meningkatkan kinerja dan ketersediaan beberapa node server sehingga dapat dilakukan load balancing pada setiap service yang dibuat.

4) Database service

Database service digunakan merupakan service untuk mengelola data center.

5) Mongo DB

Mongo DB digunakan sebagai database time series untuk data sensor.

6) Postgres SQL

Postgres SQL digunakan untuk menangani database user.

7) Gateway Service

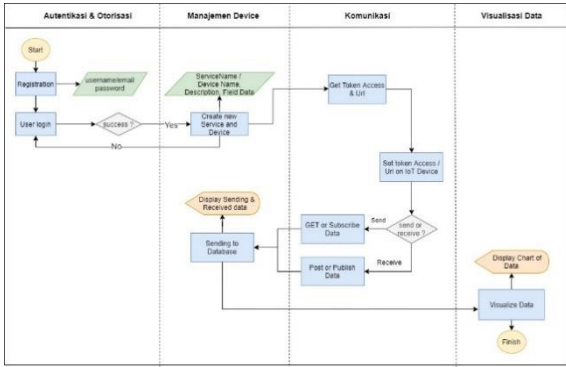
Gateway Service sebagai gerbang/ jalur konektivitas perangkat IoT dengan protokol komunikasi HTTP dan MQTT.

8) Perangkat IoT

Perangkat IoT yaitu berupa sensor dan mikrokontroler untuk mengirim data sensor dan konfigurasi.

B. Activity Diagram

Activity Diagram merupakan diagram yang menggambarkan aliran kerja atau urutan aktivitas dari sebuah sistem. Activity Diagram juga digunakan untuk mendefinisikan atau mengelompokan aluran tampilan dari sistem. Sehingga proses secara keseluruhan dapat dengan mudah dipahami sebagaimana ditunjukkan pada Gambar 3.

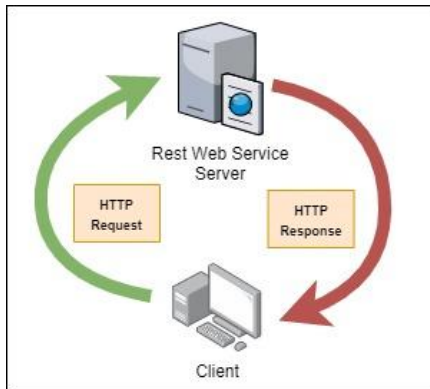


Gambar 3. Activity Diagram

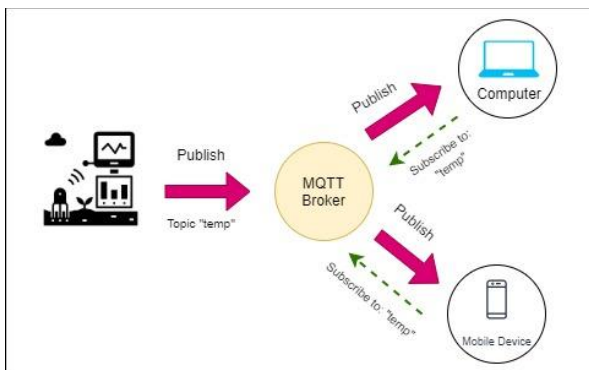
Pada diagram dapat dijelaskan beberapa proses yang diimplementasikan dalam Management Service Platform MENTOR IoT yaitu sebagai berikut:

1) Proses Komunikasi

Pada proses komunikasi ini digunakan Restfull API dan protokol komunikasi lain seperti MQTT atau socket programming. Proses ini digunakan sebagai jembatan untuk mengirimkan dan membaca data sensor dan konfigurasi pada platform IoT. Proses komunikasi yang digunakan pada platform MENTOR IoT ditunjukkan pada Gambar 4 , Gambar 5 dan Gambar 6



Gambar 4. Proses Request/Response pada Restful API

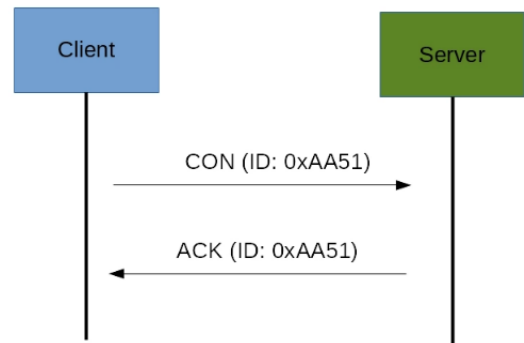


Gambar 5. Proses Publish/Subscribe pada MQTT Broker

2) Proses Autentikasi dan Otorisasi

Pada proses ini dilakukan verifikasi data user untuk mendapatkan akses pada platform IoT. Setiap user nantinya akan mendapatkan token akses setelah melakukan registrasi

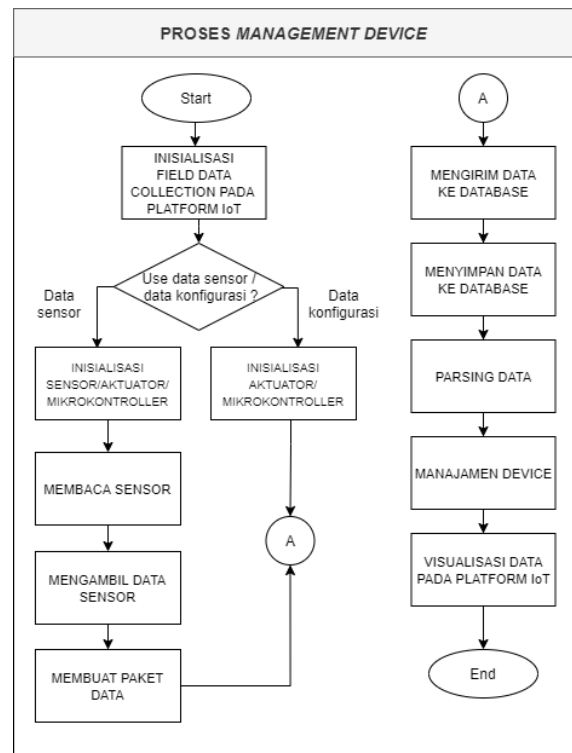
pada platform untuk mengakses service dan device yang dimiliki.



Gambar 6. Proses Request/Response pada CoAP

3) Proses Management Device

Proses ini menangani perangkat-perangkat IoT yang ada melalui mekanisme penyimpanan data dan pengaksesan data melalui protokol komunikasi yang digunakan sehingga data-data tersebut dapat mempengaruhi perangkat yang sedang berjalan. Proses management device dapat dilihat pada Gambar 7.



Gambar 7. Proses Management Device

Gambaran dari alur sistem yang dibuat yaitu sistem dapat melakukan inialisasi data service dan device. Kemudian sistem membaca data sensor dan konfigurasi kemudian membuat paket data.

Setelah paket data telah dibuat kemudian dikirim dan disimpan ke database. Kemudian data yang tersimpan

tersebut di parsing agar setiap data sensor yang dikirim dapat dikelola dan divisualisasikan.

4) Proses Visualisasi Data

Dimana pada proses ini menyediakan antarmuka pengguna untuk melihat data sensor maupun manajemen perangkat. Teknologi yang memungkinkan untuk digunakan dalam visualisasi data yaitu seperti GRAFANA, dan Chart.js.

IV. HASIL DAN PEMBAHASAN

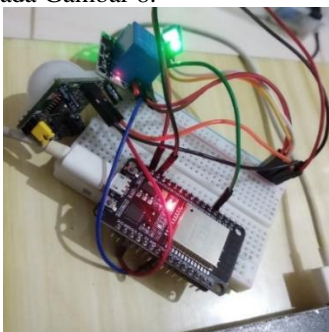
Berdasarkan perancangan yang telah dibuat, maka telah dilakukan implementasi serta pengujian platform yang meliputi pengujian fungsionalitas untuk mengetahui seberapa besar tingkat keberhasilan platform dan pengujian kinerja untuk mengetahui efektifitas platform dalam mengolah data transaksi pengguna.

A. Implementasi

Implementasi dilakukan terdiri atas 2 bagian, yaitu implementasi perangkat keras dan implementasi perangkat lunak.

1) Implementasi Perangkat Keras

Implementasi perangkat keras meliputi pengujian perangkat yang telah dibuat, yaitu sensor node 1, sensor node 2, hingga sensor node 4. Pada setiap sensor node akan mengirimkan data yang sama. Perangkat yang digunakan adalah ESP8266 tipe 12-E, ESP 32, WeMos D1 dan Arduino Uno. Implementasi ini dilakukan dengan dua metode yaitu melakukan request pengiriman dan pengambilan data melalui REST API (HTTP) dari URL yang didapatkan dari platform dan publish/subscribe data melalui Broker MQTT dari topic yang juga didapatkan dari platform. Berikut salah satu sensor node yang telah dibuat yang akan mengirimkan data sensor pada platform ditunjukkan pada Gambar 8.



Gambar 8. Sensor Node menggunakan ESP32

2) Implementasi Software

Implementasi software bertujuan untuk menguji proses halaman antarmuka pada Platform IoT yang telah dibuat mulai dari halaman registrasi dan login pada Gambar 9, halaman transaksi data pada Gambar 10 hingga halaman monitoring perangkat yang ditunjukkan pada Gambar 11.

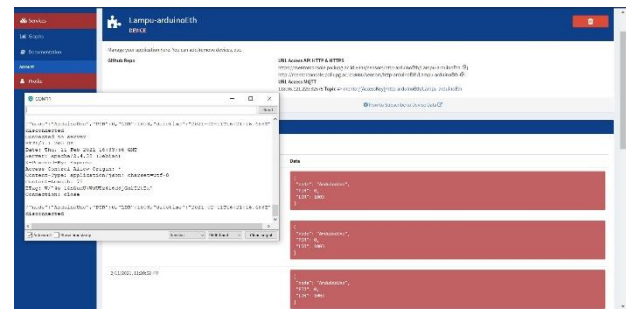
B. Pengujian Fungsionalitas

Pengujian fungsionalitas meliputi pengujian monitoring data dan perangkat IoT melalui protokol HTTP, MQTT dan

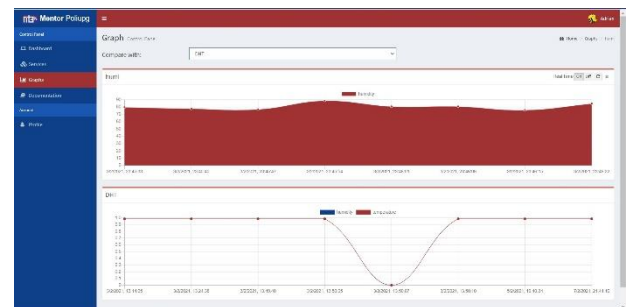
CoAP dengan menggunakan *URI access* dari setiap protokol, serta pengujian dengan melakukan kontrol pada perangkat IoT. Pengujian dilakukan pada salah satu akun pengguna yang sudah terdaftar pada platform. *URI access* yang dapat digunakan untuk monitoring perangkat ditunjukkan pada Tabel 1 dan pengujian monitoring data serta kontrol perangkat ditunjukkan pada Gambar 12 dan Gambar 13.



Gambar 9. Halaman Registrasi dan Login



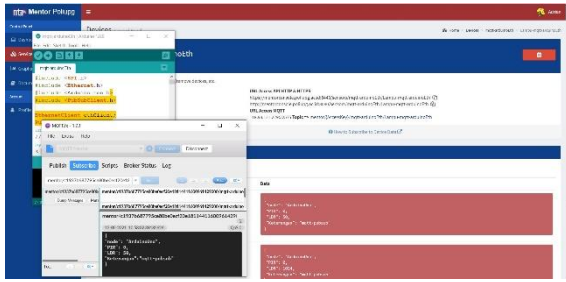
Gambar 10. Halaman Transaksi Data



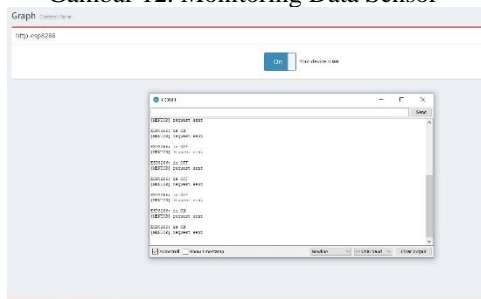
Gambar 11. Halaman Monitoring Perangkat

Tabel 1. *URI Access*

| Protokol | <i>URI Access</i> |
|----------|---|
| HTTP | HTTPS https://mentorconsole.poliupg.ac.id:8443/sensors/<Servicename>/<Devicename> HTTP http://mentorconsole.poliupg.ac.id:8080/sensors/<nama-service>/<nama-device> |
| MQTT | Host : mentorconsole.poliupg.ac.id , Port : 32575 Topic : mentor/<AccessKeyUser>/<Servicename>/<Devicename> |
| COAP | URL Endpoint coap://mentorconsole.poliupg.ac.id:30568/<AccessKeyUser>/<Servicename>/<Devicename>/{GET POST GETlast} |



Gambar 12. Monitoring Data Sensor



Gambar 13. Kontrol Perangkat

Berdasarkan pengujian fungsionalitas diatas, pengujian ini menghasilkan data pada Tabel 2.

Tabel 2. Pengujian Fungsionalitas

| Devices | Keberhasilan Transaksi Data | | | | | | Control |
|-----------------------------|-----------------------------|-----------|---------|-----------|-------|-------|---------|
| | HTTP | | MQTT | | COAP | | |
| | Req. https | Req. http | Publish | Subscribe | GET | POST | |
| ESP8266 | Ya | Ya | Ya | Ya | Ya | Ya | Ya |
| ESP32 | Ya | Ya | Ya | Ya | Ya | Ya | |
| WeMos D1 | Ya | Ya | Ya | Ya | Ya | Ya | |
| Arduino Uno Ethernet Shield | Tidak | Ya | Ya | Ya | Tidak | Tidak | |

Berdasarkan tabel 2 diatas, perangkat-perangkat IoT yang umum digunakan dapat diintegrasikan dengan platform MENTOR IoT dan berhasil melakukan transaksi data serta kontrol perangkat dengan menggunakan protokol HTTP, MQTT dan CoAP.

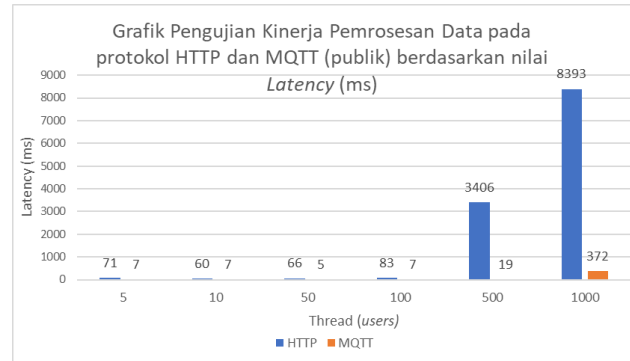
C. Pengujian Kinerja Pemrosesan Data

Pengujian kinerja dilakukan untuk mengetahui kinerja platform MENTOR IoT yang meliputi pengujian waktu pemrosesan data pada protokol HTTP dan MQTT melalui akses publik dan lokal.

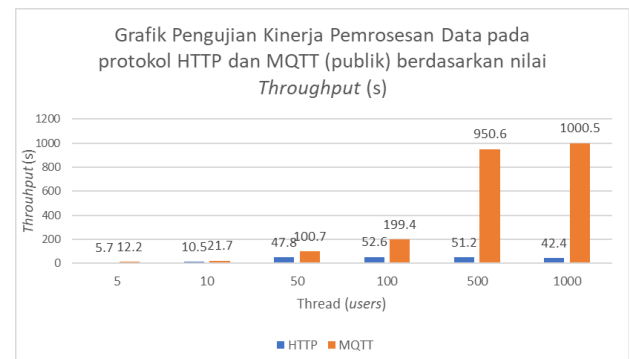
1) Pengujian Kinerja Pemrosesan Data pada protokol HTTP dan MQTT melalui akses publik

Pengujian kinerja ini dilakukan dengan melakukan request pada HTTP dan publish pada broker MQTT secara bersamaan (dapat dianalogiikan satu request sama dengan satu pengguna) melalui akses publik. Pengujian ini

menghasilkan waktu througput, latency dan persentase error saat melakukan transaksi data. Dari hasil pengujian ini diketahui bahwa waktu latency (delay) pada protokol HTTP lebih besar dibandingkan waktu latency (delay) pada protokol MQTT. sebaliknya waktu throughput pada protokol HTTP lebih kecil dibandingkan pada protokol MQTT sebagaimana ditunjukkan pada Gambar 14 dan Gambar 15.



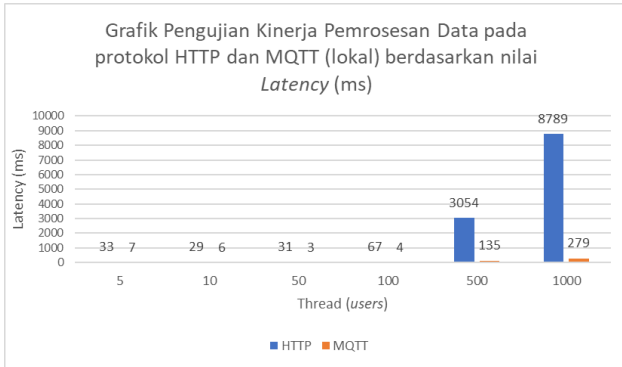
Gambar 14. Pengujian Kinerja berdasarkan nilai Latency (publik)



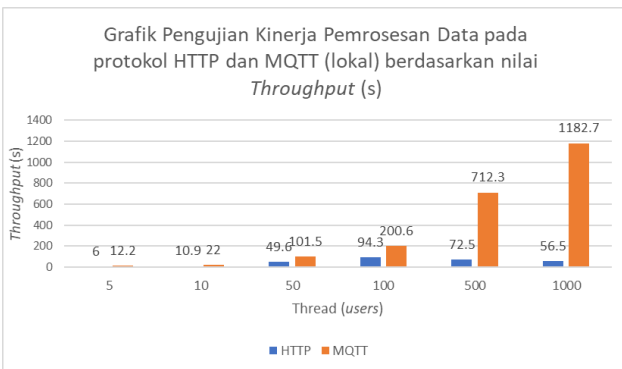
Gambar 15. Pengujian Kinerja berdasarkan nilai Throughput (publik)

2) Pengujian Kinerja Pemrosesan Data pada protokol HTTP dan MQTT melalui akses lokal

Pengujian kinerja ini dilakukan dengan melakukan request pada HTTP dan publish pada broker MQTT secara bersamaan juga dengan mengakses server lokal. Pengujian ini menghasilkan waktu througput, latency dan persentase error saat melakukan transaksi data. Dari hasil pengujian ini juga diketahui bahwa waktu latency (delay) pada protokol HTTP lebih besar dibandingkan waktu latency (delay) pada protokol MQTT. sebaliknya waktu throughput pada protokol HTTP lebih kecil dibandingkan pada protokol MQTT sebagaimana ditunjukkan pada Gambar 16 dan Gambar 17.



Gambar 16. Pengujian Kinerja berdasarkan nilai Latency (lokal)



Gambar 17. Pengujian Kinerja berdasarkan nilai Throughput (lokal)

V. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan pada penelitian ini dapat disimpulkan bahwa.

1. Platform MENTOR IoT yang dibangun dapat melakukan monitoring pada perangkat IoT menggunakan protokol komunikasi HTTP dan MQTT serta dapat melakukan kontrol pada setiap perangkat IoT yang dimiliki.
2. Hasil pengujian menunjukkan bahwa platform MENTOR IoT mampu berjalan dengan berbagai protokol yang berbeda, melakukan kontrol perangkat dan dilihat dari banyaknya pengguna yang mengirimkan data secara bersamaan dan terintegrasi serta waktu yang dihasilkan saat mengirimkan data melalui akses lokal dan akses publik tidak jauh beda. Sehingga berdasarkan hasil tersebut menandakan bahwa platform ini sangat reliable dan scalable.

REFERENSI

- [1] G. C. Fox, S. Kamburugamuve, and R. D. Hartman, "Architecture and measured characteristics of a cloud based internet of things," *Proc. 2012 Int. Conf. Collab. Technol. Syst. CTS 2012*, pp. 6–12, 2012, doi: 10.1109/CTS.2012.6261020.
- [2] A. Singh and Y. Viniotis, "Resource allocation for IoT applications in cloud environments," *2017 Int. Conf. Comput. Netw. Commun. ICNC 2017*, pp. 719–723, 2017, doi: 10.1109/ICCNC.2017.7876218.

- [3] J. Zhou *et al.*, "CloudThings: A common architecture for integrating the Internet of Things with Cloud Computing," *Proc. 2013 IEEE 17th Int. Conf. Comput. Support. Coop. Work Des. CSCWD 2013*, pp. 651–657, 2013, doi: 10.1109/CSCWD.2013.6581037.
- [4] Anisa, N. (2019). Jenis IoT Network Protocols. Diakses 5 Juli 2020, IoT Network Protocols: <https://sis.binus.ac.id/2019/11/15/jenis-iot-network-protocols>.
- [5] M. Wahyu, I. Santosa, R. Primananda, and W. Yahya, "Implementasi Load Balancing Server Basis Data Pada Virtualisasi Berbasis Kontainer," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 12, pp. 6908–6914, 2018.
- [6] S. Wahyuni, "Rancang Bangun Perangkat Lunak Pada Semi Otomatis Alat Tenun Selendang Songket Palembang Berbasis Mikrokontroler ATMEGA 128," Politeknik Negeri Sriwijaya, 2015.
- [7] A. Abdurrazaq and D. Sitompul, "Jurnal - Sensor dan Pengaplikasiannya," *Sensor*, 2017.
- [8] Michael S. Davies. (2019). The Evolution of IoT Sensor Data. Diakses 4 Juli 2020, IoT Sensor Data: <https://www.iotcommunications.com/blog/iot-sensor-data>.
- [9] D. S. Ebert, J. M. Favre, and R. Peikert, "Data visualization," *Comput. Graph.*, vol. 26, no. 2, pp. 207–208, 2002, doi: 10.1016/S0097-8493(02)00051-1.
- [10] I. Logre, S. Mosser, P. Collet, and M. Riveill, "Sensor data visualisation: A composition-based approach to support domain variability," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8569 LNCS, no. July, pp. 101–116, 2014, doi: 10.1007/978-3-319-09195-2_7.
- [11] J. B. Song, J. H. Song, J. P. Chai, and Y. Yang, "Data Visualization of the Advertising Exchange Flow Statements Generation System on Demand Side Platform Based on Highcharts," *MATEC Web Conf.*, vol. 63, p. 05029, 2016, doi: 10.1051/mateconf/20166305029.